

Verify and improve your targets with Modern CMake

John Franklin Rickard | john@moduleworks.com

Example Setup

✓ include

Ⓞ hello_there.hpp

✓ source

Ⓞ hello_there_impl.cpp

Ⓞ hello_there_impl.hpp

Ⓞ hello_there.cpp

Ⓜ CMakeLists.txt

Example C++ Files

```
1 // hello_there.hpp
2 #pragma once
3
4 void hello_there();
```

Example C++ Files

```
1 // hello_there.hpp
2 #pragma once
3
4 void hello_there();
```

```
1 // hello_there.cpp
2 #include "hello_there.hpp"
3
4 #include "hello_there_impl.hpp"
5
6 void hello_there()
7 {
8     hello_there_impl();
9 }
```

Example C++ Files

```
1 // hello_there_impl.hpp
2 #pragma once
3
4 void hello_there_impl();
```

Example C++ Files

```
1 // hello_there_impl.hpp
2 #pragma once
3
4 void hello_there_impl();
```

```
1 // hello_there_impl.cpp
2 #include "hello_there_impl.hpp"
3
4 #include <iostream>
5
6 void hello_there_impl()
7 {
8     std::cout << "Hello there!" << std::endl;
9 }
```

Example C++ Files

```
1 // hello_there_impl.hpp
2 #pragma once
3
4 void hello_there_impl();
```

```
1 // hello_there_impl.cpp
2 #include "hello_there_impl.hpp"
3
4 #include <print>
5
6 void hello_there_impl()
7 {
8     std::println("Hello there!");
9 }
```

Minimal CMakeLists.txt

```
1 cmake_minimum_required(VERSION 3.24)
2 project(example1 LANGUAGES CXX)
3
4 add_library(sample
5     source/hello_there.cpp
6     source/hello_there_impl.cpp)
```


Minimal CMakeLists.txt

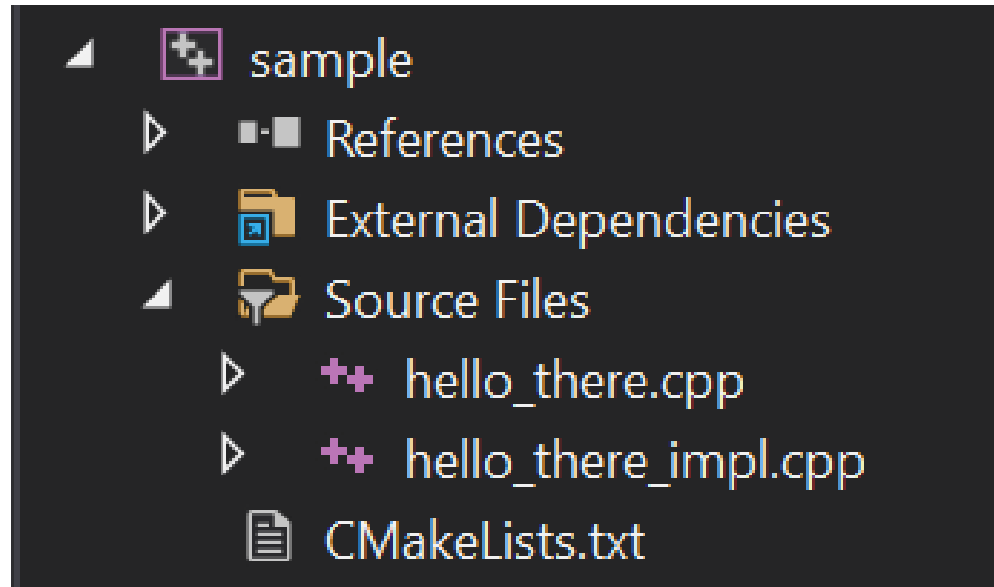
```
1 cmake_minimum_required(VERSION 3.24)
2 project(example1 LANGUAGES CXX)
3
4 add_library(sample
5     source/hello_there.cpp
6     source/hello_there_impl.cpp)
```

fatal error C1083: Cannot open include
file: 'hello_there.hpp': No such file or
directory

Minimal CMakeLists.txt

```
1 cmake_minimum_required(VERSION 3.24)
2 project(example1 LANGUAGES CXX)
3
4 add_library(sample
5     source/hello_there.cpp
6     source/hello_there_impl.cpp)
7
8 target_include_directories(sample PUBLIC include)
```

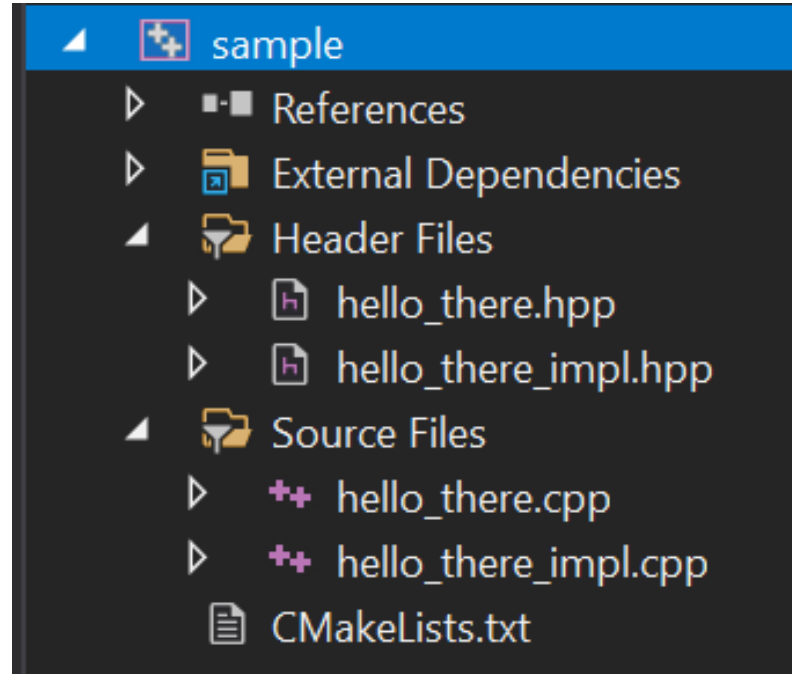
IDE Support



Minimal CMakeLists.txt

```
1  cmake_minimum_required(VERSION 3.24)
2  project(example1 LANGUAGES CXX)
3
4  add_library(sample
5      source/hello_there.cpp
6      source/hello_there_impl.cpp
7      source/hello_there_impl.hpp
8      include/hello_there.hpp)
9
10 target_include_directories(sample PUBLIC include)
```

IDE Support



Old Variable Approach

Use variable to manipulate source file list.

Often also seen in combination with globbing (bad!).

```
1 set(sample_sources
2     source/hello_there.cpp
3     source/hello_there_impl.cpp
4     source/hello_there_impl.hpp
5     include/hello_there.hpp)
6
7 add_library(sample ${sample_sources})
```

Modern Target Approach

```
1  add_library(sample)
2
3  target_sources(sample
4      PRIVATE
5          source/hello_there.cpp
6          source/hello_there_impl.cpp
7          source/hello_there_impl.hpp
8      PUBLIC
9          include/hello_there.hpp)
10
11 target_include_directories(sample PUBLIC include)
```

Added in CMake 3.11

File Sets: Usage

```
1  add_library(sample)
2
3  target_sources(sample
4      PRIVATE
5          source/hello_there.cpp
6          source/hello_there_impl.cpp
7          source/hello_there_impl.hpp
8      PUBLIC
9          FILE_SET HEADERS
10         BASE_DIRS include
11         FILES include/hello_there.hpp)
```

Added in CMake 3.23

File Sets: Syntax

```
target_sources (<target>
  [<INTERFACE | PUBLIC | PRIVATE>
  [FILE_SET <set>
    [TYPE <type>]
    [BASE_DIRS <dirs>...]
    [FILES <files>...]
  ]...
]...)
```

Required argument TYPE:

- HEADERS
- CXX_MODULES

File Set Properties

```
get_target_property(<variable> <target> <property>)
```

- HEADER_SETS
- INTERFACE_HEADER_SETS
- HEADER_SET
- HEADER_SET_<NAME>
- HEADER_DIRS
- HEADER_DIRS_<NAME>

Verify target interfaces

```
1 cmake -S . -B build
2     -DCMAKE_VERIFY_INTERFACE_HEADER_SETS=ON
```

- Sets initial value of target property `VERIFY_INTERFACE_HEADER_SETS`
- Generates test targets for each CMake library with public File Sets

Added in CMake 3.24

Generated test targets

```
1 cmake --build build
2     --target all_verify_interface_header_sets
```

<target_name>_verify_interface_header_sets
per target and one source file per public header.

```
1 // hello_there.hpp.cxx
2 #include <hello_there.hpp> // IWYU pragma: associated
```

Excluding specific headers

```
1 set_source_files_properties(include/hello_there.hpp PROPERTIES
2     SKIP_LINTING ON
3 )
```

Beware, “empty” file sets dont generate anything!

No file sets to validate, no

`all_verify_interface_header_sets` generated.

Installing targets

```
1 install(TARGETS sample
2     EXPORT sample_export
3     ...)
```

Installing targets

```
1 install(TARGETS sample
2     EXPORT sample_export
3     ...)
4
5 install(
6     FILES
7     include/hello_there.hpp
8     DESTINATION include)
```

Installing targets

```
1 install(TARGETS sample
2         EXPORT sample_export
3         ...)
4
5 install(
6     FILES
7         include/hello_there.hpp
8     DESTINATION include)
```

```
1 CMake Error at CMakeLists.txt:16 (install):
2   install TARGETS target sample is exported but not
3   all of its interface file sets are installed
```


Installing targets

```
1 install(TARGETS sample
2     EXPORT sample_export
3     ...
4     FILE_SET HEADERS DESTINATION include)
```

Thank you for your attention

Bonus slide: Spot the error

```
1 set(installFolder include)
2
3 install(TARGETS sample
4         EXPORT sample_export
5         ...
6         FILE_SET HEADERS DESTINATION ${installfolder})
```

Bonus slide: Spot the error

```
1 set(installFolder include)
2
3 install(TARGETS sample
4         EXPORT sample_export
5         ...
6         FILE_SET HEADERS DESTINATION ${installfolder})
```

```
1 cmake -S . -B build --warn-uninitialized -Werror=dev
2 ...
3 CMake Error (dev) at CMakeLists.txt:18 (install):
4     uninitialized variable 'installfolder'
```

